5-2010

A Singular-Value-Based Semi-Fragile Watermarking Scheme for Image Content Authentication with Tampering Localization

Xing Xin

Utah State University

A SINGULAR-VALUE-BASED SEMI-FRAGILE WATERMARKING SCHEME FOR IMAGE CONTENT AUTHENTICATION WITH TAMPERING LOCALIZATION

by

Xing Xin

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

_____	_____
Xiaojun Qi Major Professor	Stephen W. Clyde Committee Member
_____	_____
Vicki H. Allan Committee Member	Byron R. Burnham Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2010

ABSTRACT

A Singular-Value-Based Semi-Fragile Watermarking Scheme for
Image Content Authentication with Tampering Localization

by

Xing Xin, Master of Science

Utah State University, 2010

Major Professor: Dr. Xiaojun Qi
Department: Computer Science

This thesis presents a novel singular-value-based semi-fragile watermarking scheme for image content authentication with tampering localization. The proposed scheme first generates a secured watermark bit sequence by performing a logical "xor" operation on a content-based watermark and content-independent watermark, wherein the content-based watermark is generated by a singular-value-based watermark bit sequence that represents intrinsic algebraic image properties, and the content-independent watermark is generated by a private-key-based random watermark bit sequence. It next embeds the secure watermark in the approximation subband of each non-overlapping 4×4 block using the adaptive quantization method to generate the watermarked image. The image content authentication process starts with regenerating the secured watermark bit sequence following the same process mentioned in the secured watermark bit sequence generation. It then extracts a possibly embedded watermark using the parity of the quantization

results from the probe image. Next, the authentication process constructs a binary error map, whose height and width are a quarter of those of the original image, using the absolute difference between the regenerated secured watermark and the extracted watermark. It finally computes two authentication measures (i.e., M1 and M2), with M1 measuring the overall similarity between the regenerated watermark and the extracted watermark, and M2 measuring the overall clustering level of the tampered error pixels. These two authentication measures are further seamlessly integrated in the authentication process to confirm the image content and localize any possible tampered areas. The extensive experimental results show that the proposed scheme outperforms four peer schemes and is capable of identifying intentional tampering, incidental modification, and localizing tampered regions.

(63 pages)

ACKNOWLEDGMENTS

To my major professor, Dr. Xiaojun Qi, who has been a mentor and guide to me throughout the research process.

To Dr. Stephen Clyde and Dr. Vicki Allan, who reviewed my thesis and gave valuable suggestions.

To my family members, who continually give me their love and support.

Xing Xin

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Significance

Trustworthy digital multimedia plays an important role in applications, such as news reporting, intelligence information gathering, criminal investigation, security surveillance, and health care. However, all too often this trustworthiness can no longer be taken for granted since users can easily manipulate, modify, or forge digital content without causing noticeable traces, using low-cost and easy-to-use digital multimedia editing software. Therefore, digital multimedia authentication has become an important issue.

Recently, digital watermarking techniques have been considered as one of the most promising techniques for multimedia authentication. The goal of watermarking is to embed into the image data a mark that can identify the copyright owner of the work. Among these, semi-fragile watermarking techniques have been proposed to protect copyright and prove tampering of the digital content. These techniques allow acceptable content-preserving manipulations, such as common image processing and JPEG/JPEG2000 compression, while detecting content-altering malicious manipulations such as removal, addition, and modification of objects.

Background

Here, I briefly review the history of digital watermarking techniques and its six important properties followed by a discussion the general framework of semi-fragile watermarking techniques and some representative semi-fragile watermarking techniques.

Digital Watermarking

Digital watermarking is a label applied to digital media to automatically detect and possibly prosecute copyright infringement. Digital watermarking is not a new technique. Its history can be traced back to 13th century Europe. At that time, a visible personal mark or signature was superimposed on an image that needed protection [1]. It is a simple but very effective method that is still widely used as a security protection method nowadays.

In the early years of digital watermarking history, despite its visibility, it worked well. Visible watermarking is clearly not ideal for art work, since superimposed marks bring distortions that decrease the visual quality. In general, an efficient digital watermarking requires the following properties [2]:

1. Invisibility: The watermark should be embedded into the image, video, or audio signal and not be visible to the user. The minimum requirement of invisibility is to keep the distortion introduced by the watermark lower than the just-noticeable distortion (JND) of the image. Several researchers have invented JND based on the contrast sensitivity function (CSF) and Watson model [3, 4].
2. Tamper detection: Watermark detection results of the existence of certain watermark information should be very reliable. This is related to two concepts, false positive alarm and false negative alarm. A false positive error happens when there is no watermark in the host media, though the detector declares there is. On the other hand, a false negative error happens when there is watermark in the host media, though the detector declares there is not.

3. Discrimination of incidental distortion and malicious tampering: The most important and difficult issue in digital watermarking, this discrimination includes tolerance to common image processing (i.e., image enhancement, and median filtering) and image compressions (i.e., JPEG, and JPEG2000). A semi-fragile watermarking scheme is supposed to be able to survive all those distortions but still detect malicious tampering (i.e., adding, removing, and changing objects).
4. Security: The embedded watermark should be impervious to forgery and manipulation.
5. Identification of tampered areas: The location of altered areas should be highly disposed to estimation and the other areas highly disposed to verification as authenticated.
6. Oblivion with no transmission of any secret information: The original image or explicit information derived from the original image should not be needed in the authentication process.

Semi-Fragile Watermarking

Figure 1 shows the framework of a typical semi-fragile watermarking scheme. It consists of two components: the embedding scheme and the extraction scheme. Here, I briefly review these two components.

The Embedding Scheme. The embedding scheme consists of two steps: watermark generation and watermark embedding.

The watermark generation step generates the watermark(s) to be embedded. In general, the watermark can be a randomly generated binary sequence, a binary image, or

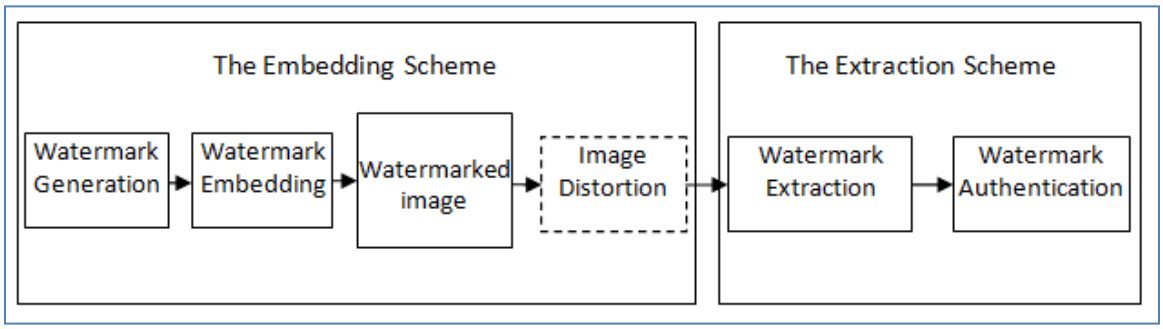


Figure 1. The framework of a typical semi-fragile watermarking scheme.

a content-based signature. A content-based signature can be obtained by feature extraction techniques.

The watermark embedding step embeds the generated watermark message into the original image. A variety of watermark embedding techniques have been proposed in the literature. These techniques can be categorized into spatial domain-based and frequency domain-based embedding techniques. In general, frequency-based watermarking techniques are better than spatial-based watermarking techniques from the following two perspectives:

1. Frequency-based watermarking techniques can achieve better invisibility than spatial-based watermarking techniques because a small modification of some of the coefficients in the frequency domain causes small global changes when transforming back to the spatial domain (i.e., all the coefficients in the transformed spatial domain are changed on a small scale).
2. Frequency-based watermarking techniques are more robust than spatial-based watermarking techniques because the relationship of coefficients in the frequency

domain cannot be easily affected by attacks which modify the coefficients in the spatial domain.

Most watermarking techniques use either an additive or a multiplicative function in the frequency domain, i.e., discrete Fourier transform (DFT), discrete cosine transform (DCT), or discrete wavelet transform (DWT) to embed watermark information. Both types of functions keep the least significant bits or the parity of transformation coefficients, or the relationship of certain transformation coefficients.

The Extraction Scheme. The extraction scheme consists of two steps: watermark extraction and watermark authentication.

The watermark extraction step should be specifically designed to pair with the embedding scheme to retrieve the embedded watermark under various intentional or un-intentional attacks that may occur in the real world.

The watermark authentication step compares the extracted watermark with the embedded watermark to authenticate the image content. For watermarked images that undergo some incidental distortions, there will be little or no difference between the extracted and the embedded watermarks. That is, this slight difference can be used to indicate that watermarks can be successfully extracted under incidental distortions. In other words, the scheme is robust to incidental distortions. For the watermarked images that undergo malicious attacks, the extracted watermark will be significantly different from the embedded watermark. That is, this significant difference can be used to validate the authenticity of the image content and to localize the distortion areas if malicious

attacks do take place. In other words, the scheme is fragile to malicious tampering distortions.

Next, I briefly review several representative semi-fragile watermarking schemes in the domain of DCT or DWT. In general, these schemes use the chosen transform domain as the media to embed and extract watermarks. They then use the extracted watermarks to authenticate the digital content and localize the tampered areas if possible.

DCT-based Semi-fragile Watermarking Schemes. Lin et al. [5] propose embedding Gaussian distribution-based block patterns in the DCT domain. Tampering detection is accomplished by verifying the correlation on these block patterns. This scheme can identify altered regions within a watermarked image with 75% accuracy under moderate compression and near 90% accuracy under light compression. Lin and Chang [6] propose to generate the invariant features at a predetermined JPEG quality factor and embed these features into mid-frequency of 8×8 DCT blocks. This scheme is robust against substitution of blocks and improves on the method proposed in [5], in that false alarms near edges hardly occur. However, it fails to detect malicious attacks that preserve the sign of the DCT coefficients. Ho and Li [7] propose a similar yet better scheme by using the relationship of DCT coefficients in low and middle frequencies. This scheme protects the authenticity of a compressed watermarked image while the JPEG quality is higher than the authors' predefined lowest authenticable quality. Maeno et al. [8] propose two methods to address the shortcomings of [6]. The first method adds a random bias factor to the fixed decision boundary to catch the malicious manipulation and keep the false alarm rate low. The second method uses a non-uniform quantization

scheme to improve accuracy in encoding the relationships between paired transform coefficients and increase the alteration detection sensitivity.

DWT-based Semi-fragile Watermarking Schemes. Kundar and Hatzinakos [9] embed a watermark in a quantized DWT domain. Zhou et al. [10] propose to embed a signature from the original image into the wavelet coefficients. Kang and Park [11] incorporate the just noticeable differences feature to improve the performance of [5] for discriminating malicious from nonmalicious attacks. Hu and Han [12] propose to extract image features from low-frequency wavelet coefficients to generate two watermarks: one for classifying the intentional content modification and the other for indicating the modification location. Liu et al. [13] use Zernike moments in the DWT domain as features for the authentication task. Zhu et al. [14] apply the block-mean-based quantization strategy to embed the inter-block and intra-block signatures in the DWT domain for tamper detection and localization, respectively. Yang and Sun [15] embed the watermark by integrating the human visual system model to modify the vertical and horizontal subbands of image subblocks. Che et al. [16] use the dynamic quantized approach to embed watermark in low-frequency wavelet coefficients. Cruz et al. [17] employ the vector quantization method to embed a robust signature into the approximation subband of each image sub-block. However, all these schemes are only robust to moderate JPEG compression (i.e., JPEG compression of higher than a 50% to 60% quality factor). The false alarm rates for watermarking schemes proposed in [9, 10, 11, 12, 13, 14, 17] are high under common image processing attacks. Specifically, the schemes proposed in [9] and [13] achieve a 32×32 detection unit, and Cruz's scheme [17] achieves a 16×16 detection unit.

Organization of Thesis

In this paper, we propose a novel semi-fragile watermarking scheme by generating a secure watermark that results from performing the logical operation "*xor*" between a content-based watermark and content-independent watermark. Here, a content-based watermark is a singular-value-based feature, and a content-independent watermark is a private-key-based random watermark. The proposed scheme then embeds the secure watermark in the wavelet domain using the adaptive quantization method. The proposed watermarking scheme further utilizes two authentication measures derived from a binary error map to authenticate the image content and localize the tampered areas. This scheme also possesses all the desired properties mentioned earlier for an effective authentication watermarking scheme.

The remainder of this thesis is organized as follows. Chapter II presents the proposed scheme, Chapter III analyzes the performance of the proposed scheme, Chapter IV presents the extensive experimental results of the proposed scheme, and Chapter V summarizes the conclusions and provides directions for future work.

CHAPTER II

THE PROPOSED APPROACH

The proposed semi-fragile watermarking scheme consists of four components: secured watermark generation, watermark embedding, watermark extraction, and watermark authentication. This chapter starts with a brief introduction of several important notations and concepts used in my thesis, followed by a detailed explanation of each component.

Important Notations and Concepts

In this section, I briefly introduce the singular value decomposition (SVD), DWT, and the terminology used in the following sections for ease of discussion of the proposed approach.

SVD

Any $m \times n$ real-valued matrix A with m >= n can be written as the product of three matrices: $A = USV^T$. The columns of the $m \times m$ matrix U are mutually orthogonal unit vectors, as are the columns of the $n \times n$ matrix V. The $m \times n$ matrix S is a pseudo-diagonal matrix, and the diagonal entries are known as SVs (Singular Values) of A. While both U and V are not unique, the SVs are fully determined by A. That is, the SVs of a matrix are unique. The SVs of a square matrix of size $n \times n$ are n descending values along the diagonal of the matrix S. From the viewpoint of image processing applications, SVs represent intrinsic algebraic image properties. Figure 2 shows a simple example of

$$A = USV^T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}$$

$$\text{where } A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

Figure 2. Example of singular value decomposition.

singular value decomposition. The two singular values are 5 and 3, which are located along the diagonal of the matrix S.

DWT

DWT includes many kinds of transforms, such as Haar wavelet, Daubechies wavelet, and others. My thesis utilizes the Haar wavelet. For an input represented by a list of 2^n numbers, the Haar wavelet transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale: finally resulting in $2^n - 1$ differences and one final sum. For ease of understanding, Figure 3 shows the workflow of DWT. After applying a 1-level DWT on an image, we get the approximation subband *LL*, the horizontal subband *LH*, the vertical subband *HL*, and the diagonal subband *HH*. Moreover, if we want to apply a 2-level DWT on the image, we just simply apply another 1-level DWT on the approximation subband *LL*. After applying a 2-level DWT, we also get the approximation subband *LL2*, the horizontal subband *LH2*, the vertical subband *HL2*, and the diagonal subband *HH2* of the approximation subband *LL* other than subbands *LH*, *HL*, *HH*.

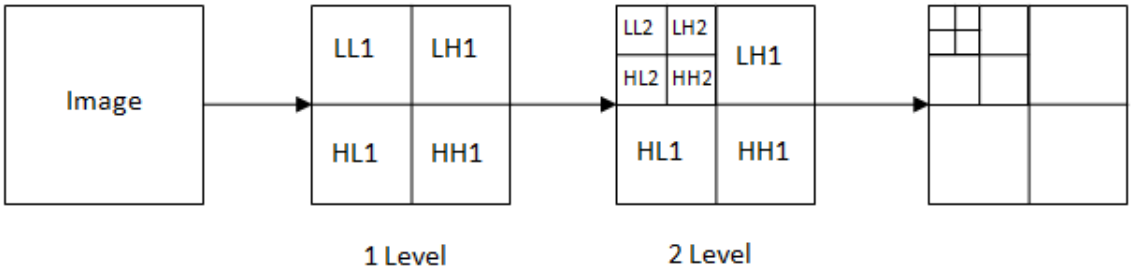


Figure 3. The workflow of discrete wavelet transform.

Table 1 Watermarking Terminology.

Host image/Original image	The image used for embedding the watermark
Probe image	The image used for watermark extraction
W and *W'*	The original and extracted watermark sequences
I and *I'*	The original and watermarked images
k	The secret key

Terminology

For the purposes of this discussion, it is useful to summarize the most commonly used terminology. Various research groups, such as image processing, communication theory, and cryptography, have studied watermarking. Although, they have slightly different terminology from each other, most of these approaches share some common standards or rules. Table 1 provides a guide to the most frequently used terminology.

Secured Watermark Generation

Using the specific relationships of SVs of the horizontal, vertical, and diagonal subbands of each 4×4 block of the original image, we generate a content-based watermark that represents intrinsic algebraic image properties to facilitate the

authentication process. The detailed steps for generating the content-based watermark are as follows:

1. Divide the original image *I* into non-overlapping 8×8 blocks.
2. For each 8×8 block (i.e., *Blk*), modify its coefficient (i.e., *Blk(x,y)*) to an integral multiple (i.e., *modified-Blk(x,y)*) of the quantization matrix *Q* (shown in Figure 4) which is used in JPEG compression [18], where 1 <= *x* <= 8 and 1 <= *y* <= 8. See formula (1). After applying this on each block, the original image is represented as a quantized image *modified-I*.

$$\text{modified} - Blk(x,y) = round(Blk(x,y)/Q(x,y)) \times Q(x,y) \quad (1)$$

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Figure 4. Quantization matrix.

3. Divide the quantized image *modified-I* into non-overlapping 4×4 blocks.
4. For each 4×4 block *modified-*B_i, where *i* ranges from 1 to the total number of blocks, perform the following operations:

   4.1.Divide into 2×2 sub-blocks to obtain *subblock1*$_i$, *subblock2*$_i$, and *subblock3*$_i$ (shown in Figure 5). The upper left *subblock* is not used. Here, I briefly

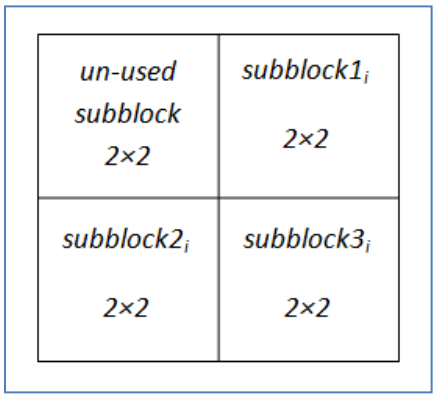


Figure 5. Illustration of subblocks.

explain why it is not used. As discussed in the next section, Watermark Embedding, we choose to use the upper-left value *X* of the approximation subband *LL* of *modified-B*$_i$ as the media for the embedding process. *X* is exactly related to the four values in the *un-used subblock*. So, we choose not to use this upper left *sub-block* to ensure the robustness of both the content-based watermark and the embedding scheme.

4.2. Apply SVD on *subblock1*$_i$ to obtain three matrices *U1*$_i$, *S1*$_i$, and *V1*$_i$, where *subblock1*$_i$ = *U1*$_i$ ×*S1*$_i$ ×*V1*$_i^T$. Apply the SVD on *subblock2*$_i$ and *subblock3*$_i$ to obtain *U2*$_i$, *S2*$_i$, and *V2*$_i$, and *U3*$_i$, *S3*$_i$, and *V3*$_i$, respectively.

4.3. Generate a watermark bit based on the relationship of SVs of *subblock1*$_i$, *subblock2*$_i$, and *subblock3*$_i$. These SVs correspond to the three values (i.e., *S1*$_i$ (1, 1), *S2*$_i$ (1, 1), and *S3*$_i$ (1, 1)). The singular values are in descending order as introduced above, so we choose the first one, which is the most notable and stable, to generate the watermark bit. The rules for generating the content-based watermark bit *CW*$_i$ are as follows:

4.3.1. Generate bit *B*$_1$ using *S1*$_i$ (1,1) and *S2*$_i$ (1, 1) based on the relationship:

$$B_1 = \begin{cases} 1 & \text{if } S1_i(1,1) >= S2_i(1,1) \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

4.3.2. Generate bit B_2 using $S2_i$ (1,1) and $S3_i$ (1, 1) based on the relationship:

$$B_2 = \begin{cases} 1 & \text{if } S2_i(1,1) >= S3_i(1,1) \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

4.3.3. Generate bit B_3 using $S3_i$ (1,1) and $S1_i$ (1, 1) based on the relationship:

$$B_3 = \begin{cases} 1 & \text{if } S3_i(1,1) >= S1_i(1,1) \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

4.3.4. Generate the content-based watermark bit CW_i using:

$$CW_i = xor(xor(B_1, B_2), B_3) \quad (5)$$

For security reasons, we also generate a random content-independent watermark bit sequence *IW*, which has the same length as the content-based watermark. This watermark bit sequence is generated by using the Mersenne Twister algorithm [19] and a private key *k* [6]. Using the correct private key, the same *IW* can be generated for the original image and its probe images (i.e., possibly distorted watermarked images).

The embedded secured watermark W_i is finally generated by performing the logical operation "*xor*" on the random watermark IW_i and the content-based watermark CW_i.

Watermark Embedding

We divide the original image into non-overlapping 4×4 blocks and sequentially embed *W* in the wavelet domain of each 4×4 block. We utilize the parity of the quantized value of the approximation subband to embed the watermark. To ensure the watermark's invisibility and increase robustness against common image processing attacks, we choose to use the upper-left value *X* of the approximation subband as the media for the

embedding process. The strategy of embedding a watermark bit is as follows. Compute the quantized value X_q by getting the integer part of X divided by a quantizer q. If the parity of X_q equals to the embedding bit, change X to $X_q \times q$. Otherwise, change X to $X_q \times q$ plus q. All these changes ensure that the parity of the modified X is consistent with the embedding bit. It should be noted that the bigger q is, the bigger the changes, consequently, the worse the quality of the watermarked image, and the stronger the robustness. In our system, the value of q is adaptive and different for each block. Specifically, as introduced in the section entitled Secured Watermark Generation, we use the total of B_1, B_2, and B_3 (e.g., *Sum*) of each block to decide the corresponding quantization value of q.

$$q = \begin{cases} 11 & \text{if } B_1 + B_2 + B_3 = 0 \\ 13 & \text{if } B_1 + B_2 + B_3 = 1 \\ 15 & \text{if } B_1 + B_2 + B_3 = 2 \\ 17 & \text{if } B_1 + B_2 + B_3 = 3 \end{cases} \tag{6}$$

The detailed embedding procedure is shown below. It should be noted that the boundary check process summarized in step 2.5 is necessary when some blocks are all 0's (black) or all 255's (white).

1. Divide the original image I into non-overlapping 4×4 blocks.
2. For each 4×4 block B_i and its corresponding embedded watermark bit W_i, perform the following operations:
   2.1 Apply the 1-level Haar wavelet transform to obtain the approximation subband LL_i, the horizontal subband LH_i, the vertical subband HL_i, and the diagonal subband HH_i.

2.2 Quantize the upper-left value of LL_i (i.e., $LL_i(1,1)$) by its quantizer q, as computed by equation (3),using:

$$X_q = \lfloor LL_i(1,1)/q \rfloor \tag{7}$$

2.3 Modify the $LL_i(1,1)$ value by:

$$LL_i(1,1) = \begin{cases} X_q \times q & \text{if } \bmod(X_q,2) == W_i \\ X_q \times q + q & \text{otherwise} \end{cases} \tag{8}$$

where mod(X_q, 2) computes the remainder of X_q divided by 2.

2.4 Apply the inverse 1-level Haar wavelet transform to obtain the watermarked block.

2.5 Perform the boundary check on the 2×2 upper-left corner of the watermarked block to ensure that its four values are in the proper range. For an 8-bit grayscale image, this range is [0- q/4, 255+q/4]. If any of the four values in the 2×2 upper-left corner falls outside of the proper range, apply the following remedy strategies:

a) If one value is larger than the upper-bound of the allowable range, modify $LL_i(1,1)$ by:

$$LL_i(1,1) = X_q \times q - q \tag{9}$$

b) If one value is smaller than the lower-bound of the allowable range, modify $LL_i(1,1)$ by:

$$LL_i(1,1) = X_q \times q + 2 \times q \tag{10}$$

c) Apply the inverse 1-level Haar wavelet transform to obtain the corrected watermarked block. If any value in the 2×2 upper-left corner of the

corrected watermark block falls outside of the proper range, modify its value by adding or subtracting $4 \times q$ to ensure the modified value is in the proper range and the parity of X_q is intact.

Figure 6 illustrates the effects of embedding the watermark in the wavelet domain, using the above quantization method. This figure shows that each $LL_i(1,1)$'s is modified to the nearest 0 bin (the dashed line) or 1 bin (the solid line) according to its quantized value X_q and the embedding bit W_i.

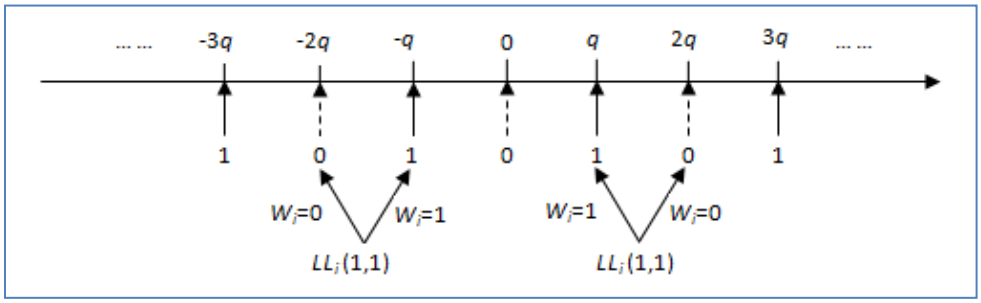


Figure 6. Illustration of the quantization process.

Watermark Extraction

The watermark extraction process uses the same blocking strategy to divide the image into non-overlapping 4×4 blocks. It then uses the parity of the quantized upper-left value X' of the approximation subband of each block to extract the watermark bit. Here, the q value used for quantization is calculated by using the same strategy as in the watermark embedding process. The detailed steps are as follows:

1. Divide the probe image I' into non-overlapping 4×4 blocks.
2. For each 4×4 block B_i', perform the following operations:

2.1 Apply the 1-level Haar wavelet transform to obtain the approximation subband LL_i', the horizontal subband LH_i', the vertical subband HL_i', and the diagonal subband HH_i'.

2.2 Quantize the upper-left value of LL_i' (i.e., LL_i'(1,1)) by its quantizer q computed by equation (5) using:

$$X'_q = round(LL'_i(1,1)/q) \tag{11}$$

2.3 Set the extracted watermarked bit EW_i as mod(X_q', 2).

Watermark Authentication

We generate a binary error map to perform the watermark authentication task. First of all, we simulate the process of generating the secured watermark *W'* by applying the logical operation "*xor*" on the content-based watermark *CW'* and the content-independent random watermark *IW'*. Here, *CW'* is a regenerated content-based watermark using the same strategy introduced in the Secured Watermark Generation section, and *IW'* is a content-independent watermark that is exactly the same as *IW* introduced in the Secured Watermark Generation section. Since the extracted watermark *EW* reflects the changes of local intensity resulting from attacks, we construct the error map, i.e., *ErrorMap*, by mapping the absolute difference between EW_i and W'_i (i.e., $|EW_i - W'_i|$) onto its corresponding 4×4 block. The 0's and 1's in *ErrorMap* indicate the match and mismatch between extracted and embedded watermarks, respectively. In other words, any pixel with the value of 1's in *ErrorMap* is an error pixel. In the proposed system, we classify the error pixels into three categories: strongly tampered, mildly tampered, and isolated error pixels. Figure 7 illustrates these three categories of error pixels in red solid circles

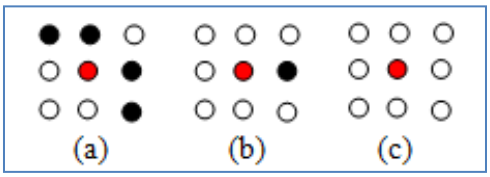


Figure 7. Examples of three categories of error pixels shown in red solid circles. (a) strongly tampered error pixel, (b) mildly tampered error pixel, and (c) isolated error pixel.

using a window size of 3×3. Specifically, we consider an error pixel as strongly tampered if at least four of its eight neighbors are error pixels (marked by black solid circles); an error pixel as mildly tampered if one, two, or three of its eight neighbors are error pixels; and an error pixel as isolated (i.e., likely caused by noise) if none of its eight neighbors is an error pixel. As a result, we do not consider the isolated error pixel as the tampered error pixel and consider both strongly tampered and mildly tampered error pixels as tampered error pixels. It should be noted that the window size and thresholds of the number of neighboring error pixels in the window for defining strongly tampered and mildly tampered error pixels can be set differently based on the specific application requirement. They also determine the sensitivity of the authentication process.

We next define two authentication measures, M_1 and M_2, to protect copyright and prove tampering, where M_1 measures the overall similarity between extracted and embedded watermarks and M_2 measures the overall clustering level of the tampered error pixels. We compute M_1 as the percentage of error pixels (i.e., 1's) in *ErrorMap*. We compute M_2 as the ratio between the number of strongly tampered error pixels and the number of tampered error pixels in *ErrorMap*. The detailed steps for computing M_2 are as follows:

1. Initialize *TamPixNum* and *StrongTamPixNum* as 0's, where *TamPixNum* stores the number of tampered error pixels and *StrongTamPixNum* stores the number of strongly tampered error pixels.
2. For each error pixel, perform the following operations:

   2.1. If it is tampered (i.e., mildly or strongly tampered), add *TamPixNum* by 1.

   2.2. If it is strongly tampered, add *StrongTamPixNum* by 1.
3. Compute M_2 by:

$$M_2 = \begin{cases} 0 & \text{if } TamPixNum = 0 \\ \dfrac{StrongTamPixNum}{TamPixNum} & \text{otherwise} \end{cases} \tag{12}$$

Finally, we design a quantitative method to decide the authenticity of the probe image based on the two authentication measures. The algorithmic view of the authentication process is summarized below:

1. Compute M_1 using *ErrorMap.*
2. If $M_1 \leq T_{median}$ (i.e., 0.15), update *ErrorMap* as its 3×3 median filtering result.
3. Compute M_2 using *ErrorMap.*
4. If $0 \leq M_1 < T_{halferrorbit}$,
   a. if $M_2 < T_{malicious}$, the probe image is authenticated
   b. else the probe image is maliciously attacked.
5. If $T_{halferrorbit} \leq M_1 < T_{errorbit}$.
   a. if M2 < Tmalicious, the probe image is incidentally attacked
   b. else the probe image is maliciously attacked.
6. If $M_1 \geq T_{errorbit}$, the probe image is not embedded with watermarks.

It is important to apply the median filtering on *ErrorMap* when M_1 is less than or equal to 0.15 (i.e., at most 15% of 4×4 blocks are detected as distorted). Due to the small amount of distortions, we can infer that the probe image must have undergone small malicious attacks or moderate incidental attacks. That is, the tampered regions would be small and tend to cluster if malicious attacks occurred, and the tampered regions would be small and tend to scatter if moderate incidental attacks occurred. This median filtering removes all mildly tampered error pixels. It also treats non-error pixels as error pixels if the non-error pixels are surrounded by at least five error pixels. That is, this filtering keeps the clustered error pixels intact and makes scattered mildly tampered error pixels and isolated error pixels disappear. As a result, the small malicious attack leads to a larger M_2 value due to the removal of mildly distorted error pixels. The extensive experiments show that the value of 0.15 for T_{median} works well on all 30 test images and all the simulated attacks.

The remaining thresholds, i.e., $T_{errorbit}$, $T_{halferrorbit}$, and $T_{malicious}$, involved in the authentication process are determined based on the predefined false negative probability of 10^{-6}. The threshold for $T_{errorbit}$ is derived as follows. The probability for a pixel in *ErrorMap* to be detected as 0 or 1 is 0.5. So, each pixel is a binomially distributed random variable. The expected value (i.e., $E(errorbit)$) and the variance of error bits (i.e., $Var(errorbit)$) are respectively $0.5 \times numel$ and $0.5 \times (1\text{-}0.5) \times numel = 0.25 \times numel$, where *numel* is the total number of pixels in *ErrorMap*. Therefore, we deduce the threshold for detecting if the probe image has been embedded with the watermark bits by:

$$P(T_{errorbit} \geq \tau_1) = 1 - P(T_{errorbit} < \tau_1) \approx 1 - \Phi\left[\frac{\tau - E(errorbit)}{\sqrt{Var(errorbit)}}\right] \geq 1 - 10^{-6}$$
$$\Rightarrow \tau_1 \geq 7924.9 \Rightarrow T_{errorbit} \geq \frac{7924.9}{numel} = 0.4837 \quad (13)$$

Here, Φ approaches the normal distribution with an expected value of 0 and a variance of 1 when *numel* is large. Hence, we consider that the image is not embedded with the watermark if $M_1 \geq T_{errorbit} = 0.4837$ and use a half of $T_{errorbit}$ as $T_{halferrorbit}$, which is the threshold for distinguishing the incidentally attacked watermarked images from authenticated watermarked images.

The threshold for $T_{malicious}$ is derived as follows. The probability for a pixel to be detected as tampered error pixels is $½[1-(½)^8]=255/512=0.4980$. The probability for a pixel to be detected as strongly tampered error pixel is $(C_8^4 + C_8^5 + C_8^6 + C_8^7 + C_8^8) \times 0.5^9 = 163/512 = 0.3184$. Then, the expected value (i.e., *E*(*strongtampix*)) and the variance of strongly tampered error pixels (i.e., *Var*(*strongtampix*)) are $163/512 \times numel$ and $163/512 \times (1\text{-}163/512) \times numel = 0.3184 \times 0.6816 \times numel = 0.2170 \times numel$, respectively. The expected value of tampered error pixels (i.e., *E*(*tampix*)) is $0.4980 \times numel$. Therefore, we deduce the threshold for detecting malicious attacks by:

$$P(T_{malicious} \geq \tau_2) = 1 - P(T_{malicious} < \tau_2) \approx 1 - \Phi\left[\frac{\tau - E(strongtampix)}{\sqrt{Var(strongtampix)}}\right] \geq 1 - 10^{-6}$$
$$\Rightarrow \tau_2 \geq 4964.9 \Rightarrow T_{malicious} \geq \frac{4964.9}{E(tampix)} = 0.6085 \quad (14)$$

That is, we consider the attack on the watermarked image is malicious if $M_2 \geq T_{malicious} = 0.6085$.

Validation of Defined Error Pixels and Authentication Measures

The definitions of the three categories of error pixels and the two authentication measures are guided by the following observations. 1) Most error pixels would spread across the error map if incidental attacks were made on the watermarked image. 2) Most error pixels would cluster in distorted regions if malicious attacks were made on the watermarked image. Figure 8 demonstrates these two observations by showing the error pixel distribution after performing no attack and performing three attacks (i.e., obvious malicious attack by adding a black square, JPEG compression attack with the 80% quality factor, and obvious malicious attack by adding a black square followed by JPEG compression of an 80% quality factor) on the standard watermarked "Lena" image, respectively. For the error pixel distribution under each attack, we sequentially display the distribution of all error pixels (i.e., *ErrorMap*), tampered error pixels, and strongly tampered error pixels. We clearly observe the following. 1) Figure 8(a) shows that *ErrorMap* contains all 0's when no attack occurs to the watermarked image. In other words, all the watermark bits are successfully extracted, and the probe image is authentic. 2) Figure 8(b) shows that *ErrorMap* contains exclusively clustered tampered error pixels when the malicious attack is applied to the watermarked image. The strongly tampered error pixels are also clustered within the tampered areas under this malicious attack without any JPEG compression. 3) Figure 8(c) shows that *ErrorMap* contains a majority of randomly spread tampered error pixels when the incidental attack is applied to the watermarked image. The strongly tampered error pixels tend to be isolated under the

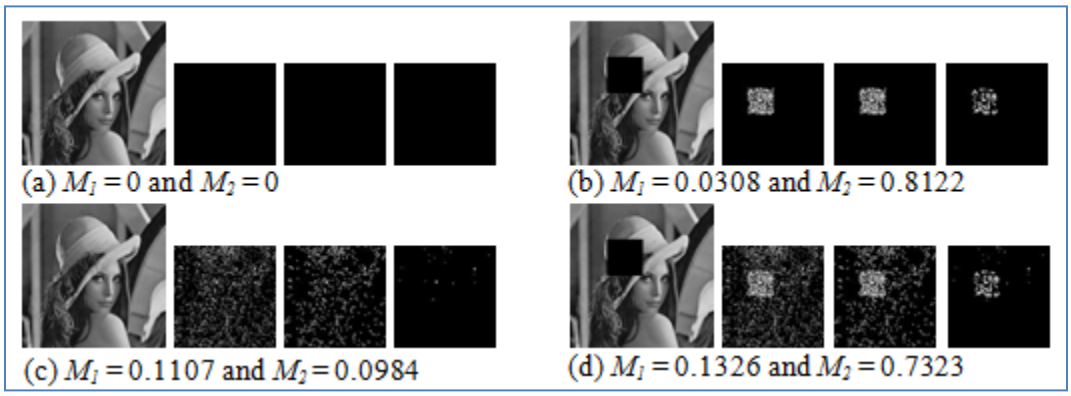


Figure 8. Illustration of the error pixel distribution.

incidental attack. 4) Figure 8(d) shows that *ErrorMap* contains a majority of clustered tampered error pixels resulting from the malicious attack, and a few randomly spread tampered error pixels resulting from the JPEG compression attack, when the combined malicious and JPEG attack is applied to the watermarked image. The strongly tampered error pixels are also clustered within the tampered areas under this combined attack. Based on the predefined thresholds, the system successfully detects each watermarked image shown in Figure 8(a), Figure 8(b), Figure 8(c), and Figure 8(d) as authenticated, maliciously attacked, incidentally attacked, and maliciously attacked, respectively.

Figure 8 illustrates the error pixel distribution: (a) the watermarked image; (b) the maliciously attacked watermarked image without any JPEG compression; (c) the 80% JPEG compressed watermarked image; (d) The maliciously attacked watermarked image followed by 80% JPEG compression, along with their corresponding *ErrorMap*'s in terms of all error pixels, tampered error pixels, and strongly tampered error pixels. The size of the error map is enlarged for easy reading of error pixels.

CHAPTER III

PERFORMANCE ANALYSIS

In the following, we quantitatively evaluate the performance of the proposed scheme in terms of the quality of the watermarked image, the robustness of the secured watermark, the tampering detection sensitivity, and the localization capability.

Quality of the Watermarked Image

In the proposed scheme, image distortion is caused by modifications of the wavelet coefficients in the embedding process. Both the quantizer q and the watermark payload p (i.e., the number of watermark bits embedded in the host image) affect the quality of the watermarked image. A larger quantizer incurs more modification to the wavelet coefficients and consequently results in more degradation of the watermarked image. Similarly, a larger payload leads to more degradation of the watermarked image. In the following, we derive the mean squared error (MSE) incurred in the embedding process, using the assumption that the original wavelet coefficients are uniformly distributed over the range of $[kq, (k+1)q]$ for $k \in Z$. When the parity of the quantization result of the original wavelet coefficient $LL_i(1, 1)$ matches the embedded watermark bit W_i, $LL_i(1,1)$ is modified to the lower-bound kq, and the MSE caused by this quantization is:

$$MSE_1 = \frac{1}{q}\int_0^q \tau^2 d\tau = \frac{q^2}{3} \tag{15}$$

Otherwise, LL_i (1, 1) is modified to the upper-bound $(k+1)q$ and the MSE caused by this quantization is:

$$MSE_2 = \frac{1}{q}\int_0^q (\tau - q)^2 d\tau = \frac{q^2}{3} \tag{16}$$

As a result, the average distortion caused by embedding one watermark bit is $q^2/3$, and the MSE of embedding p watermark bits in the block-based wavelet domain is:

$$MSE = \frac{p \times q^2}{3 \times W \times H} \tag{17}$$

where W and H are the width and the height of the host image, respectively. According to Parseval's theorem, the MSE of the entire image equals its counterpart in the wavelet domain [20]. Therefore, the *PSNR* value of the watermarked image is:

$$PSNR = 20\log_{10}\left(\frac{255}{q}\sqrt{\frac{3 \times W \times H}{p}}\right) = 20\log_{10}\left(\frac{255}{q}\sqrt{\frac{3 \times W \times H}{\frac{W \times H}{bs \times bs}}}\right) = 20\log_{10}\left(\frac{255 \times bs}{q}\sqrt{3}\right) \tag{18}$$

where *bs* denotes the size of each embedding square block. The above clearly reveals that the quality of the watermarked image is determined by both p and q. Smaller p's and q's lead to larger *PSNR* values. In the proposed system, p equals to $W \times H/16$. Based on formula (14), if a fixed quantizer is used for the quantization of watermark embedding process, the expected *PSNR* values are 44.12, 42.66, 41.42, and 40.33 for quantizers of 11, 13, 15, and 17, respectively.

The experimental results on 30 standard 8-bit grayscale images show that by using the proposed scheme with the adaptive quantizer, the average *PSNR* value of their watermarked images is 41.39. This average is consistent with the computed expected values and is higher than the empirical value (i.e., 35.00 db) for the image without perceivable degradation [21].

Robustness of the Secured Watermark

Since we use the same strategy to generate the secured watermark in both the watermark embedding process and watermark authentication process, the robustness of the secured watermark is important to the proposed scheme. In other words, ideally, the regenerated secured watermark at the extraction side is supposed to be the same as the secured watermark generated in the embedding process. To analyze the robustness of the secured watermark, we next clarify three aspects of the analysis process, which aspects are also the main steps of generating the secured watermark.

Using a specific non-overlapping 4×4 block *Blk* as an example, we start the explanation with the quantization of non-overlapping 8×8 blocks, using the quantization matrix used in JPEG compression. By applying this step, modification within the range of [-*coef's*/2, *coef's*/2] can be preserved. That is, when modifications have been applied to block *Blk*, the secured watermark bit can still be regenerated if the modification on a particular value of a block falls within the range of [-*coef's*/2, *coef's*/2], where *coef's* is the quantization value at the same position of the quantization matrix (shown in Figure 4). From Figure 4, we can tell that coefficients of different locations in block *Blk* have a different robustness since their corresponding quantization values differ at different locations.

Secondly, we calculate the SV's of *subblock1*, *subblock2*, and *subblock3* of block *Blk*. Because we generate bits B_1, B_2, and B_3 using the relationships among the SV's and the relationships are more stable than the SV's themselves, it is reasonable to assume that the regenerated watermark bit will be the same as the corresponding watermark bit in the

embedding process. In other words, B_1, B_2, and B_3, which encode the relationships between each pair of the SV's of the quantized horizontal, vertical, and diagonal subbands, will not be changed even when the values of SV's can be changed because of modifications of block *Blk*.

Thirdly, we generate the watermark bit by using formula (5) to complement any possible changes in the three relationships encoded in B_1, B_2, and B_3. In other words, this step is to increase the possibility that the regenerated watermark bit is the same as the watermark bit generated in the embedding process, even when B_1, B_2, or B_3 (i.e., either of the three relationships) is different from the ones generated in the embedding process. For example, changes in $S1_i$ (1, 1) may lead to the changes in any of two relationships, i.e., B_1 and B_3. If both relationships are changed, our watermark generation process ensures that the regenerated watermark bit sequence is the same as the watermark bit sequence generated in the embedding process. Even when either of the relationships is changed, the regenerated watermark bit sequence may still stay the same if B_2 is changed. As a result, our proposed content-based watermark generation scheme is robust when incidental attacks are applied to the watermarked image. Our extensive experimental results also confirm this.

Here, we use one detailed example to illustrate the robustness of the secured watermark.

Figure 9(a) shows generating the content-based watermark bit of a 4×4 block in the embedding process. The block is from the example host image "Lena" whose size is 512×512, and the coefficients of the block are the intersection of row 1 to 4 and column 9

to 12 of "Lena." So, the upper left 16 coefficients of quantization matrix Q are used in the modification process in Figure 9(a). Figure 9(b) shows regeneration of the content-based watermark bit of the block at the same position after applying JPEG compression on the host image. Here, we choose a compression ratio as 75, which is the default compression ratio of jpeg compression in Matlab. From Figure 9, we can see that even though the block has been changed after JPEG compression, the content-based watermark bit "*Bit*'" regenerated in the authentication process is the same as the watermark bit "*Bit*" generated in the embedding process.

Tampering Detection Sensitivity

The tampering detection sensitivity of the proposed scheme is determined by the quantizer. The error map captures the changes in the quantization results and makes the tampering detectable for $k \in Z$ in the following two cases:

1. The wavelet coefficient $LL_i'(1,1)$ of the watermarked image is $2kq$, and the manipulation causes a shift of $LL_i'(1,1)$ in the range of $[(0.5+2k)q, (1.5+2k)q)$.
2. The wavelet coefficient $LL_i'(1,1)$ of the watermarked image is $2kq+q$, and the manipulation causes a shift of $LL_i'(1,1)$ in the range of $[(1.5+2k)q, (2.5+2k)q)$.

That is, the scheme is capable of detecting all the changes satisfying the above two conditions. Small changes of a half of the quantizer q or other changes falling in the range of $[(-0.5+2k)q, (0.5+2k)q]$ in the distorted area do not modify the parity of the quantized approximation value. As a result, the scheme is robust to moderate image content preserving attacks that do not dramatically change the pixel intensity.

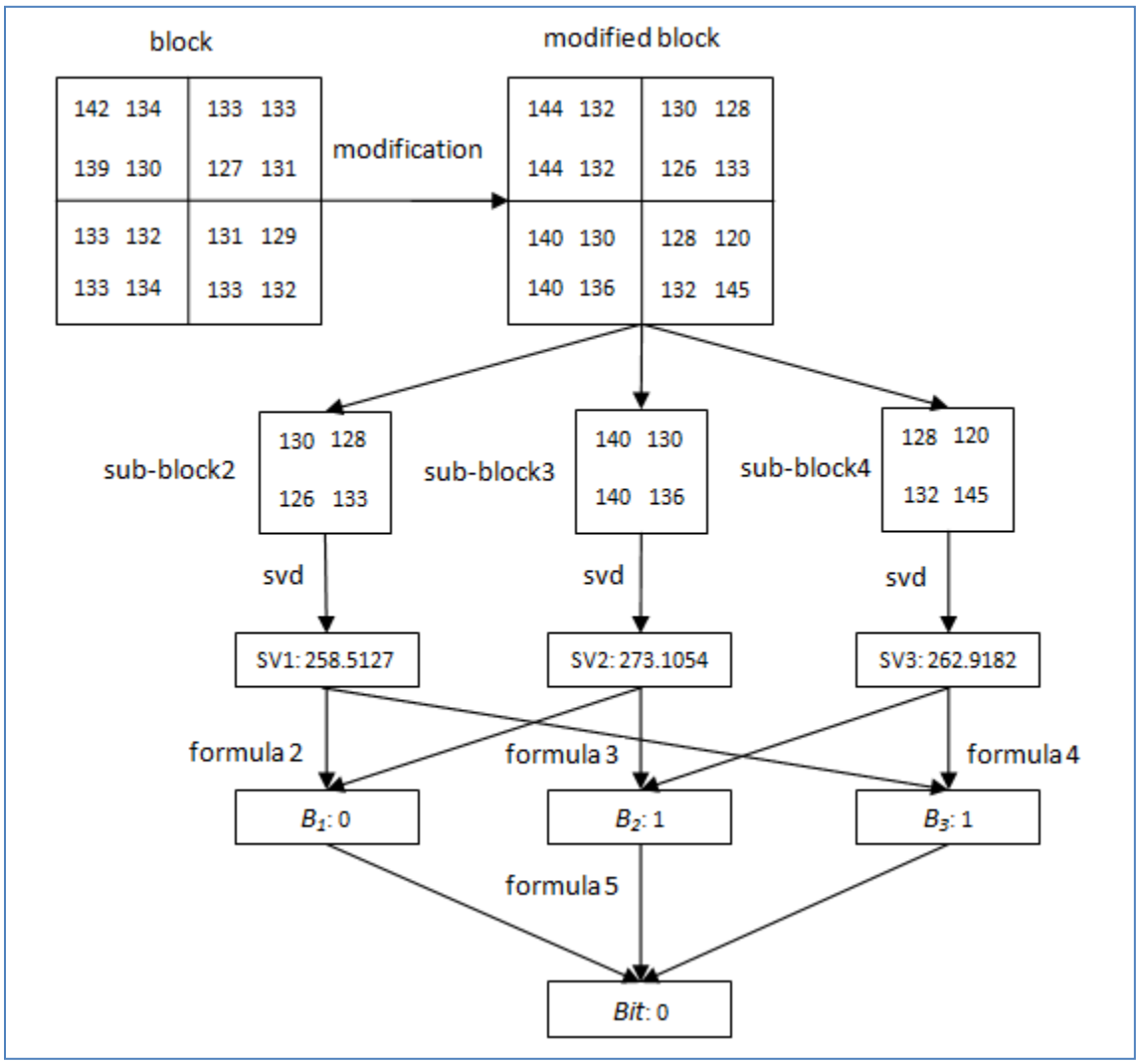


Figure 9. Example of the robustness of the secured watermark. (a) Generate the content watermark bit of the 4×4 block in embedding process.

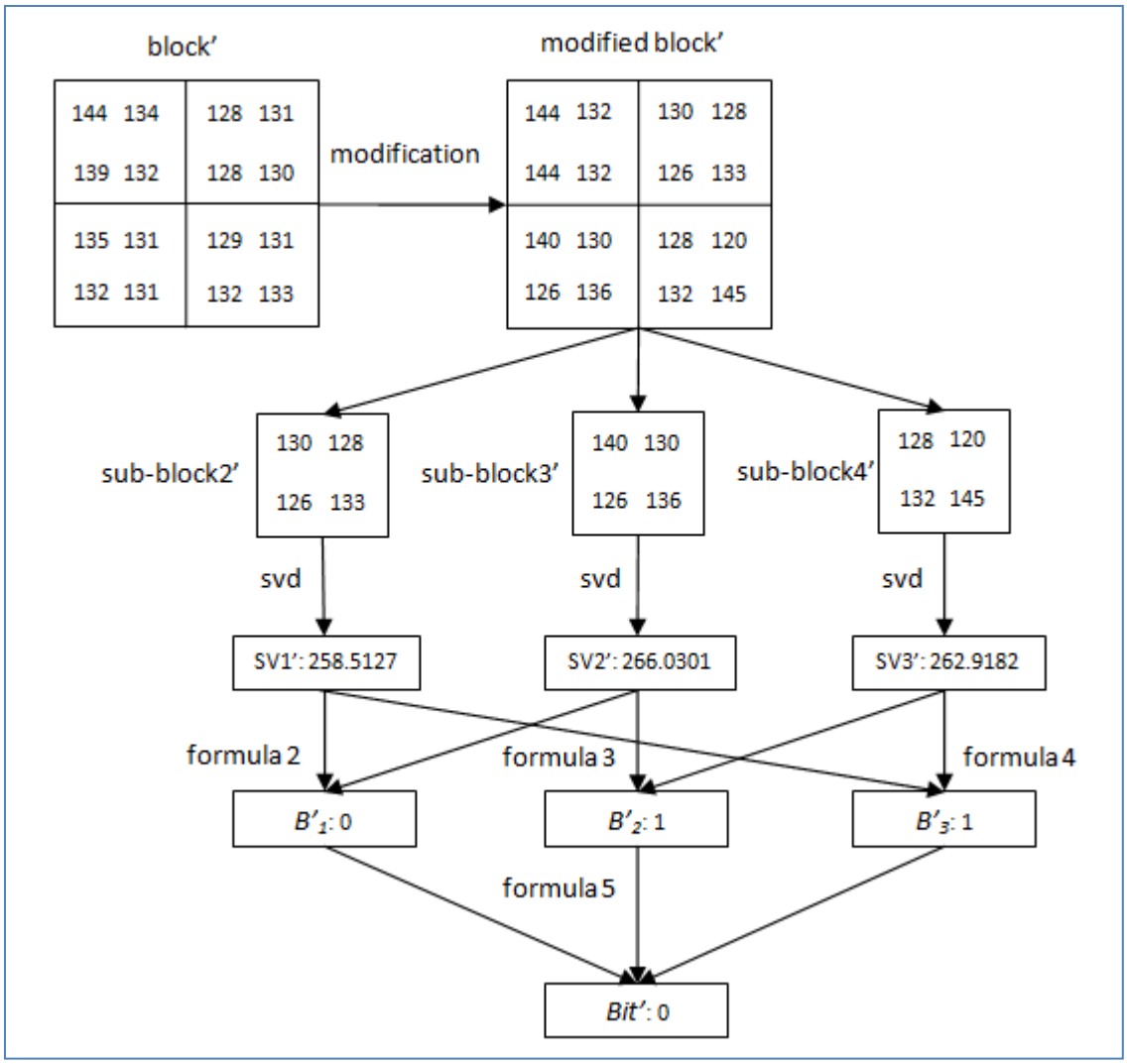


Figure 9 cont. Example of the robustness of the secured watermark.
(b) Generate the content watermark bit of the 4×4 block in authentication process.

However, some pixels in the tampered area may be missed when the changes in the wavelet domain do not satisfy the above two conditions. To address this shortcoming, the authentication process utilizes the distribution of the detected error pixels to evaluate the authenticity of the probe image. Specifically, the observations shown in Figure 8 are incorporated to compensate for the possible misclassification in *ErrorMap*. Furthermore, q is adaptive in the proposed scheme. That is, q's vary for different non-overlapping blocks in the proposed scheme. Such variation reduces even further the possibility of misclassifying, given that the range of $[(-0.5+2k)q, (0.5+2k)q]$ where the misclassification will happen is inconsistent across the whole image.

Generally, tampering detection sensitivity can still be adjusted by choosing different window sizes and thresholds. If the threshold is preset, the larger the window size, the lower the sensitivity. Based on the application requirements, the proposed scheme can identify various tampered areas and detect bigger alterations, while still bypassing smaller alterations using a predetermined window size.

Localization Capability

In the proposed scheme, image content is monitored by the embedded and extracted watermark. Specifically, changing a value in the upper-left 2×2 corner of each 4×4 block may result in a mismatch in *ErrorMap*. To compensate for the misclassification, we employ a window size of 3×3 to categorize each non-isolated error pixel as either strongly tampered or mildly tampered, as defined in the Watermark Authentication section of Chapter II. Localization capability refers to the capability to find the smallest tampered area (also termed the detection unit) in a probe image. Here, we start the

analysis with any possibly smallest 3×3 block in *ErrorMap*, wherein all nine pixels in the block are error pixels and all the pixels outside of the block are non-error pixels (shown in Figure 10(a.1)). Based on the definition of three kinds of error pixels (illustrated in Figure 7), we know that the five error pixels marked by solid red circles are strongly tampered error pixels and all nine error pixels in the 3×3 block are tampered error pixels. Therefore, $M_2 = 5/9 = 0.5556$, which is less than the threshold of $T_{malicious}$ (i.e., 0.6085), and we conclude that the 3×3 block is not a maliciously tampered area. However, the values of M_2 are 0.625 and 0.714 when the error pixel distributions follow the sample patterns shown in Figure 10(a.2) and Figure 10(a.3), respectively. That is, the proposed scheme can achieve a 12×12 detection unit when the error pixels follow the sample distributions shown in Figure 10(a.2) and Figure 10(a.3).

Next, we consider any 3×4 or 4×3 block containing 12 error pixels (shown in Figure 10(b.1)) in *ErrorMap*. All the remaining pixels in *ErrorMap* are non-error pixels. To simplify the discussion, we only consider a block of 3×4 since the authentication results for a block of 4×3 can be similarly derived. Based on the definition of three kinds of error pixels, we know that the eight error pixels marked by solid red circles are strongly tampered error pixels and all the 12 error pixels are tampered error pixels. Therefore, $M_2 = 8/12 = 0.6667$, which is larger than the threshold of $T_{malicious}$ (i.e., 0.6085), and we conclude that the 3×4 block is a maliciously tampered area. That is, the scheme can successfully achieve a 12×16 or 16×12 detection unit using the proposed authentication measures. It can also correctly identify smaller malicious attacks within the detection unit of 12×16 (i.e., the tampered areas resulting from these malicious attacks are irregular

and nonblock-based), which follow the sample error pixel distributions as shown in Figure 10(b.2) through Figure 10(b.6), since their values of M_2's are all larger than the threshold of 0.6085.

Table 2 summarizes error pixel distributions in a 3×3 block as shown inFigure 10(a), and Table 3 summarizes error pixel distributions in a 3×4 block as shown in Figure 10(b).

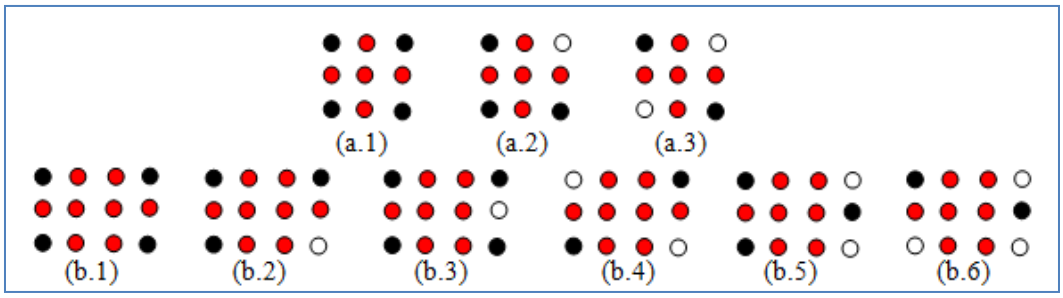


Figure 10. Illustration of different error pixels distributions in 3×3 and 3×4 blocks.

Table 2 Error Pixel Distributions (a).

figure	number of cases	error pixels	strongly tampered pixels	M2 value
a.1	1	9	5	0.56
a.2	4	8	5	0.625
a.3	2	7	5	0.714

Table 3 Error Pixel Distributions (b).

figure	number of cases	error pixels	strongly tampered pixels	M2 value
b.1	1	12	8	0.67
b.2	4	11	8	0.73
b.3	8	11	7	0.64
b.4	4	10	8	0.8
b.5	2	10	7	0.7
b.6	4	9	7	0.78

CHAPTER IV

EXPERIMENTAL RESULTS

To evaluate the performance of the proposed semi-fragile watermarking scheme, we first compared the quality of the watermarked images of the proposed scheme and four peer schemes, namely, Maeno et al.'s scheme using the random bias [6], Yang and Sun's scheme [15], Che et al.'s scheme [16], and Cruz et al.'s scheme [17], using five representative 8-bit 512×512 grayscale images. We then conducted extensive experiments on 30 standard 8-bit grayscale images by comparing the proposed system with these four peer systems. Different kinds of attempted manipulations were simulated. These simulated manipulations include the following: ten levels of image blurring, ten levels of Gaussian low-pass filtering, ten levels of median filtering, five levels of salt and pepper noise addition, ten levels of JPEG lossy compression, ten levels of JPEG2000 lossy compression, adding an irregular shape of three kinds of gray-level intensities (i.e., black, gray, and white) without compression, and using Photoshop software to paste, delete, and modify an object.

To ensure a fair comparison, we carefully studied the authentication process of each scheme to find its equivalent measure(s) to those used in the proposed scheme. We found that all four peer schemes used a measure similar to the M_1's of the proposed scheme in their authentication process. Yang's scheme also used another measure similar to the M_2's of the proposed scheme. In addition, Yang's scheme explicitly summarized the thresholds for detecting a probe image as authentic, incidentally distorted, or maliciously distorted. The other three schemes did not explicitly mention the thresholds for their

decision making. However, we could roughly infer their thresholds from their discussions. These thresholds are around 0.3 and are a little bit higher than $T_{halferrorbit}$ (i.e., 0.2418) for M_1's in the proposed scheme. All four peer schemes visually showed the error maps or the localization results without listing the values of their authentication measures. That is, they all replied on the visual inspection to show the effectiveness of their localization results. In the experiments on various malicious attacks, we showed both the values of the authentication measures and the localization results to validate the effectiveness of the proposed scheme.

Watermark Invisibility

Figure 11 summarizes the *PSNR* values after embedding watermarks in five representative images using the proposed scheme and four peer schemes, respectively. This figure clearly shows that all of the *PSNR* values are larger than 40.00 db and are comparable with the expected *PSNR* value computed in The Quality of the Watermarked Image section of Chapter III. With the exception of Cruz's scheme [17], the *PSNR* values of the proposed scheme are also higher than or comparable to the *PSNR* values of the four peer schemes that embeds watermark bits in larger blocks of 16×16.

Robustness to Common Image Processing Attacks

We performed four kinds of representative image processing attacks on 30 watermarked images. These attacks were ten levels of image blurring attacks using circular averaging filters of radii of 1.1 to 2 with an increasing step size of 0.1, ten levels of Gaussian low-pass filtering attacks using rotationally symmetric Gaussian low-pass filters of size 3×3 and standard deviation ranging from 0.1 to 1 with an increasing step

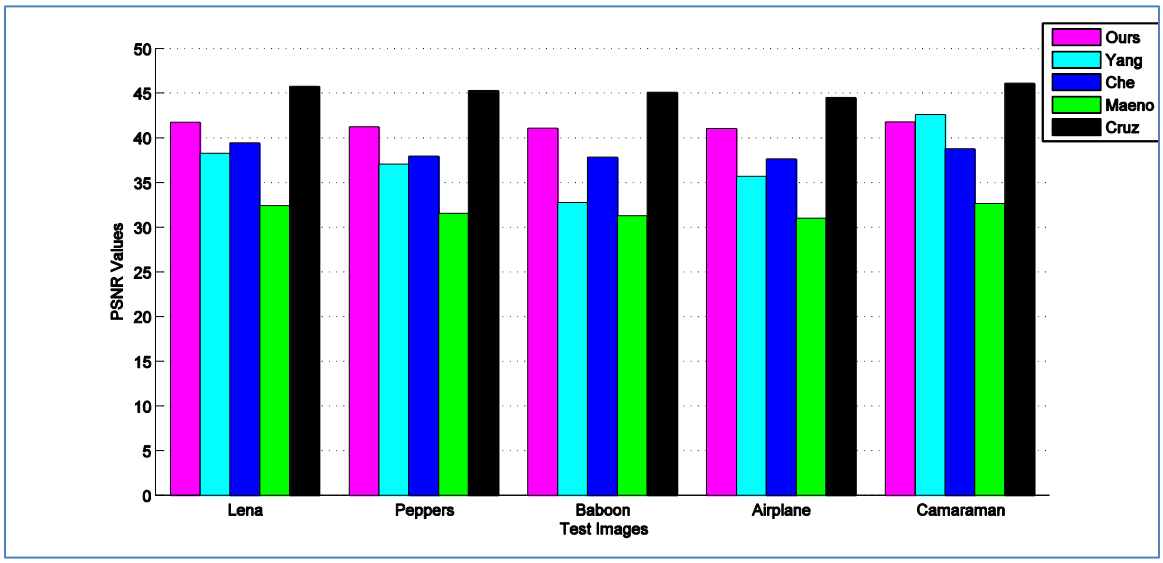
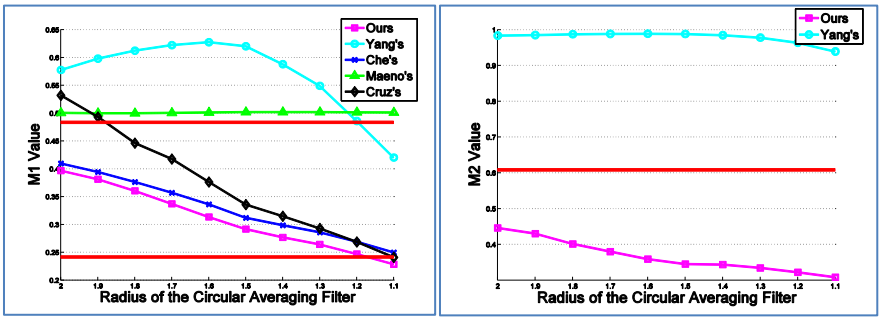


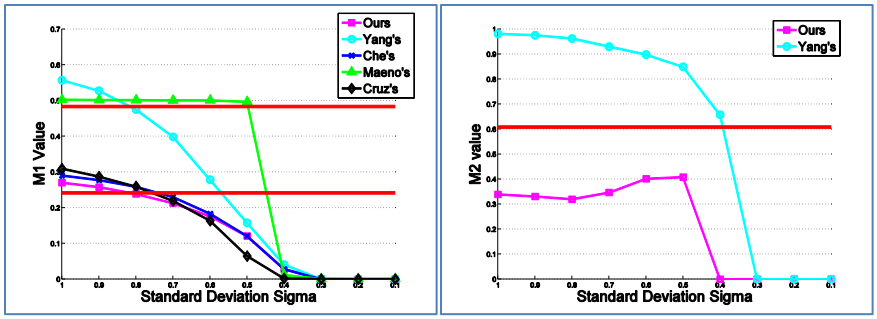
Figure 11. Comparison of PSNR values.

size of 0.1, ten levels of median filtering attacks using filters of radii of 3 to 12 with an increasing step size of 1, and five levels of salt and peppers noise attacks using noise density ranging from 0.01 to 0.05 with an increasing step size of 0.01. Since all four peer schemes used a measure similar to the M_1's of the proposed scheme in the authentication process, we plotted the average M_1 values of 30 watermarked images under each image processing attack for all five schemes on the left side of Figure 12. Yang's scheme also used another measure similar to the M_2's of the proposed scheme in the authentication process. As a result, we plotted the average M_2 values of 30 watermarked images under each image processing attack for these two schemes on the right side of Figure 12.

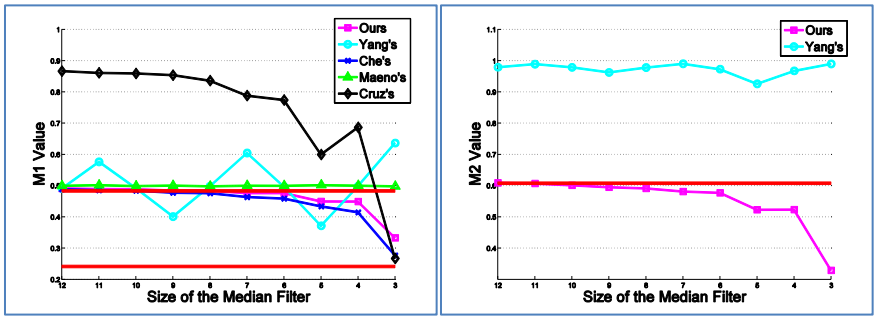
The left column in Figure 12 shows the comparison of various common image processing attacks on M_1's of the proposed scheme and four peer schemes, while the right column shows the M_2's of the proposed scheme and Yang's scheme: (a) image blurring



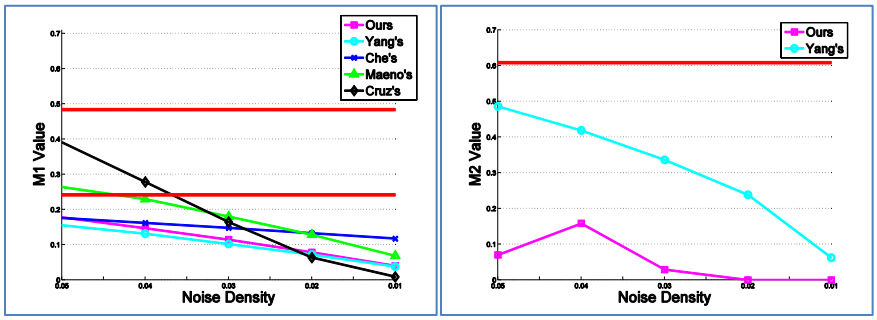
(a)



(b)



(c)



(d)

Figure 12. Comparison of various common image processing attacks.

attacks; (b) Gaussian low-pass filtering attacks; (c) median filtering attacks; (d) salt and pepper noise attacks.

Figure 12 clearly shows that all the average values of the values of M_2's of the proposed scheme are below the threshold line of 0.6085 for all image processing attacks. Specifically, the watermarked image under blurring, Gaussian low-pass filtering, or salt and pepper noise attacks were detected as authenticated if their M_1 values were smaller than $T_{halferrorbit}$ (0.2418) and as incidentally distorted if their M_1 values were between $T_{halferrorbit}$ (0.2418) and $T_{errorbit}$ (0.4837). The watermarked image under median filtering attacks with a filter size ranging from 3 to 7 was detected as incidentally distorted since its M_1 value is between $T_{halferrorbit}$ and $T_{errorbit}$. However, the scheme detected the watermarked image under median filtering attacks with a larger filter size as a non-copyrighted image since its M_1 value is larger than 0.4837. This is reasonable due to significant changes on the watermarked image. In addition, as shown in Figure 12, all average values of M_1's of the proposed scheme under all image processing attacks except the "salt and pepper" attack are the smallest among all four schemes, and all average values of M_2's of the proposed scheme are smaller than the corresponding values of Yang's method. As a matter of fact, M_1's of the proposed scheme under the salt and pepper attack are still comparable with those of Yang's scheme as shown in Figure 12(d). This indicates that under any of these image processing attacks, the proposed scheme is more robust in classifying a watermarked image as authentic or incidentally distorted. Specifically, the proposed scheme successfully detected all 30 watermarked images under

Gaussian low-pass filtering, salt and pepper noise addition, or image blurring attacks with circular averaging filters of radii smaller than or equal to 1.4 as authentic.

Robustness to JPEG Lossy Compression and JPEG2000 Lossy Compression Attacks

We performed two kinds of compressions, namely, conventional JPEG lossy compression and JPEG2000 lossy compression, on 30 watermarked images. The left plot of Figure 13 compares the average values of M_1's of 30 watermarked images under no attack and various JPEG compression attacks using quality factors of 100% down to 10% with a decreasing step size of 10% of all five schemes. The right plot of Figure 13 compares the average values of M_2's of 30 watermarked images under no attack and the same ten levels of JPEG compression attacks of the proposed scheme and Yang's scheme. Figure 13 clearly shows that all the average values of M_2's of the proposed scheme are much smaller than the corresponding average values of Yang's scheme, and they are below the threshold line of 0.6085 for JPEG compressions of a quality factor down to 10%. That is, the watermarked image under JPEG compressions of a quality factor down to 20% is detected as authentic if its M_1 value is smaller than 0.2418 and as incidentally distorted if its M_1 value is between 0.2418 and 0.4837. In addition, the average M_1 values of the proposed scheme generally are much smaller than the corresponding average values of four peer schemes for JPEG quality factors down to 30%. The proposed scheme is also the only one that increases steadily even when the quality factor is down to lower than 50%, while the others increase sharply.

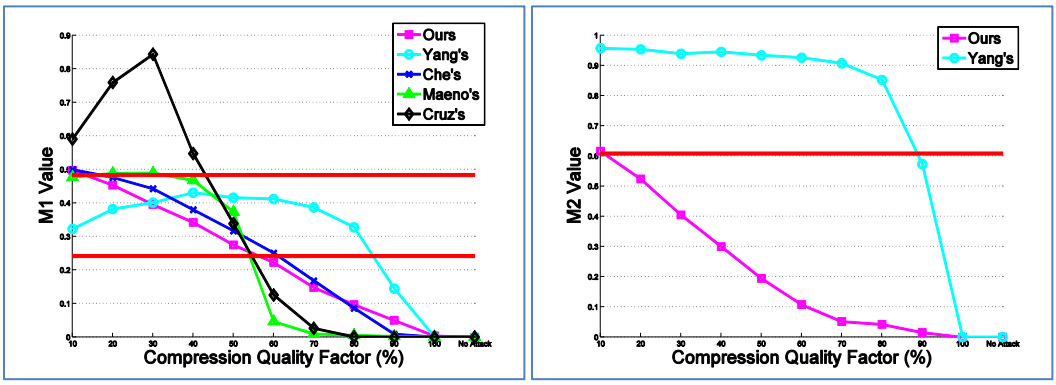


Figure 13. Comparison of various JEPG compression attacks.

All these data indicate that the proposed scheme is more robust in classifying a watermarked image under JPEG compressions of at least a 50% quality factor as authentic and classifying a watermarked image under JPEG compressions of a quality factor ranging from 10% to 50% as incidentally distorted. The experimental results on 30 watermarked images also confirm this. None of the four peer schemes achieves the comparable performance as the proposed scheme. Specifically, they detect the watermarked images under JPEG compressions of at least a 60% quality factor as incidentally distorted or authentic and detect the watermarked images under JPEG compressions of 10% to 50% quality factors as maliciously distorted.

Figure 13 also shows a comparison of various JEPG compression attacks on M_1's (left) of the proposed scheme and four peer schemes and M_2's (right) of the proposed scheme and Yang's scheme.

To evaluate the robustness of the proposed scheme to JPEG2000 lossy compression attacks, we further compare the average values of M_1's and M_2's of 30 watermarked images under various JPEG2000 compression attacks using quality factors of 1000%

down to 100% with a decreasing step size of 100%, and their equivalent JPEG compression attacks using quality factors of 100% down to 10% with a decreasing step size of 10%, as shown in Figure 14. We also plot the values of M_1's and M_2's under each attack after adding or subtracting the STDV (standard deviation values) from their average values. As clearly shown in Figure 14, all the average values of M_1's and M_2's for JPEG2000 compression attacks are much smaller than the ones for JPEG compression attacks. The relationship holds true for the average values of M_1's and M_2's adding or subtracting their corresponding STDV's. In addition, the values of M_2's are below the threshold line of 0.6085 for all JPEG2000 compressions, and the values of M_1's are below the threshold line of 0.2418 for all JPEG2000 compressions, except the one with the quality factor of 100%. That is, the watermarked image under JPEG2000 compressions of a quality factor down to 200% is detected as authentic. The experimental results also clearly demonstrate that the proposed scheme is more robust against JPEG2000 compression attacks than JPEG compression attacks since it works in the wavelet domain, which is the same domain that JPEG2000 compression works in.

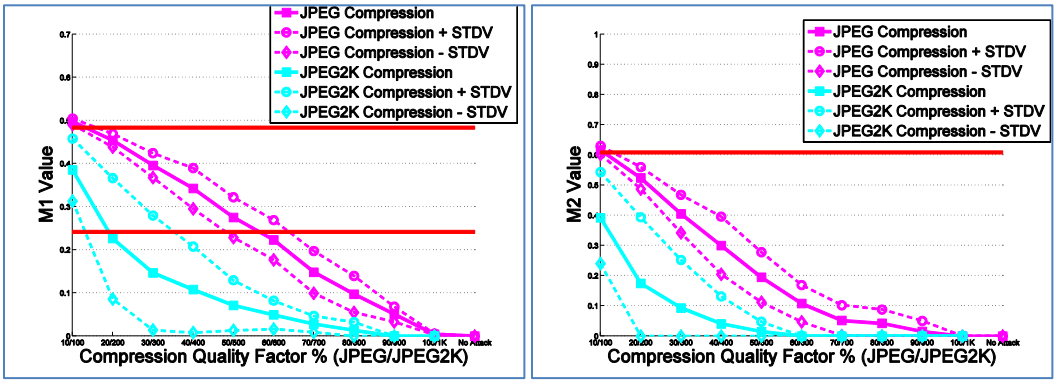


Figure 14. Comparison of JEPG2000 compression and JPEG compression attacks.

Finally, Figure 14 shows the comparison of various JEPG2000 compression attacks and their corresponding JPEG compression attacks on M_1's (left) and M_2's (right) of the proposed scheme.

Fragility to Various Malicious Attacks

We performed various malicious attacks on 30 watermarked images to demonstrate the effectiveness of the proposed scheme in localizing the maliciously tampered regions.

The yellow sections in Figure 15 show the tampering localization results of five schemes after adding an irregular shape of three kinds of gray-level intensities (i.e., black, gray, and white) to the watermarked "Lena" image, wherein black is the most dissimilar to the background intensity and gray is the most similar to the background intensity. We deliberately did not apply compression attacks to ensure that we could separate out the effect of JPEG compressions. Figure 15 clearly shows that the proposed scheme achieves similar localization results to Che's scheme and outperforms the other three schemes by correctly localizing the tampered regions regardless of the gray-level intensity of the added irregular shape. In Figure 15, whenever applicable, we also list M_1 and M_2 values in a pair for each scheme to facilitate comparison. Based on the authentication algorithm, we conclude that the proposed scheme detects these three maliciously attacked watermarked "Lena" images as maliciously tampered and correctly localizes their tampered regions. Yang's scheme is able to detect the watermarked image adding a gray or white irregular shape as maliciously attacked. However, it detects the watermarked image adding a black irregular shape as incidentally distorted. It also does not produce a decent localization result under any of the three malicious attacks. The

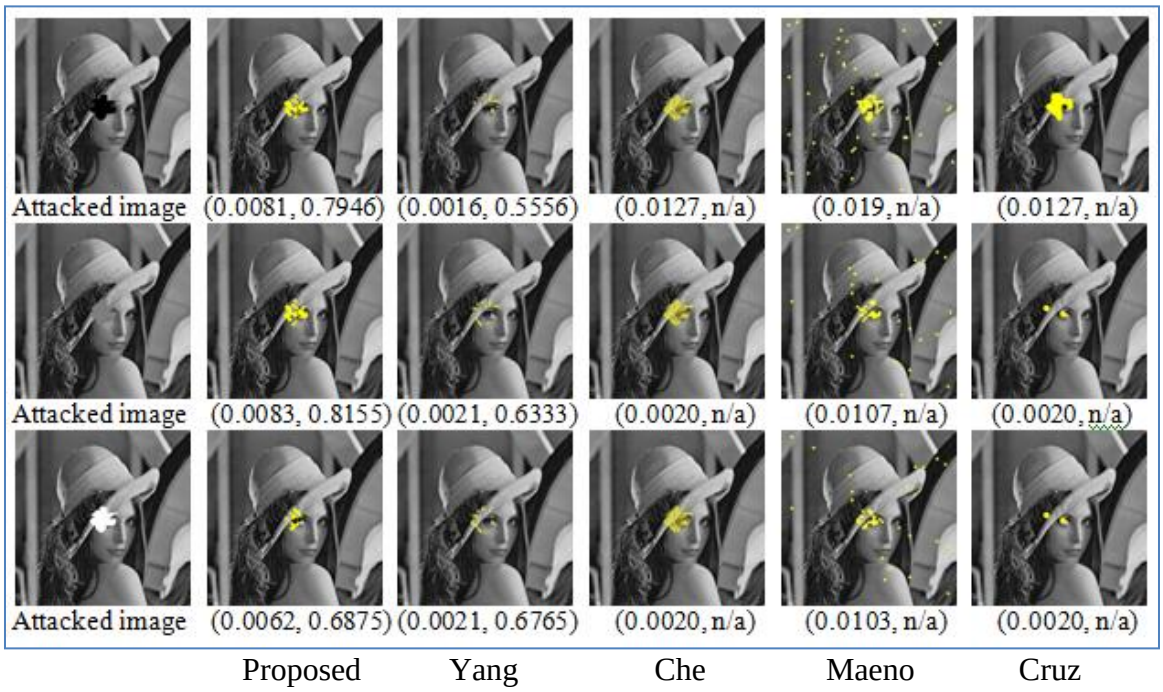


Figure 15. Comparison of malicious attacks (irregular shape).

other three peer schemes obtain small values for M_1's, which are similar to the values obtained under image processing and JPEG compression attacks. As a result, they detect these maliciously attacked watermarked "Lena" images as incidentally distorted based on the equivalent predefined thresholds.

Figure 15 also shows a comparison of the localization results after adding an irregular shape of different intensities (black, gray, and white) without JPEG compression among the following (from left to right): the proposed scheme, Yang, Che, Maeno, and Cruz.

Figure 16 demonstrates the tampering localization results on four additional representative watermarked images, which were maliciously attacked by adding the same irregular shape of three kinds of gray-level intensities (i.e., black, gray, and white) to the watermarked images. We also list the M_1 and M_2 values in a pair below each localization

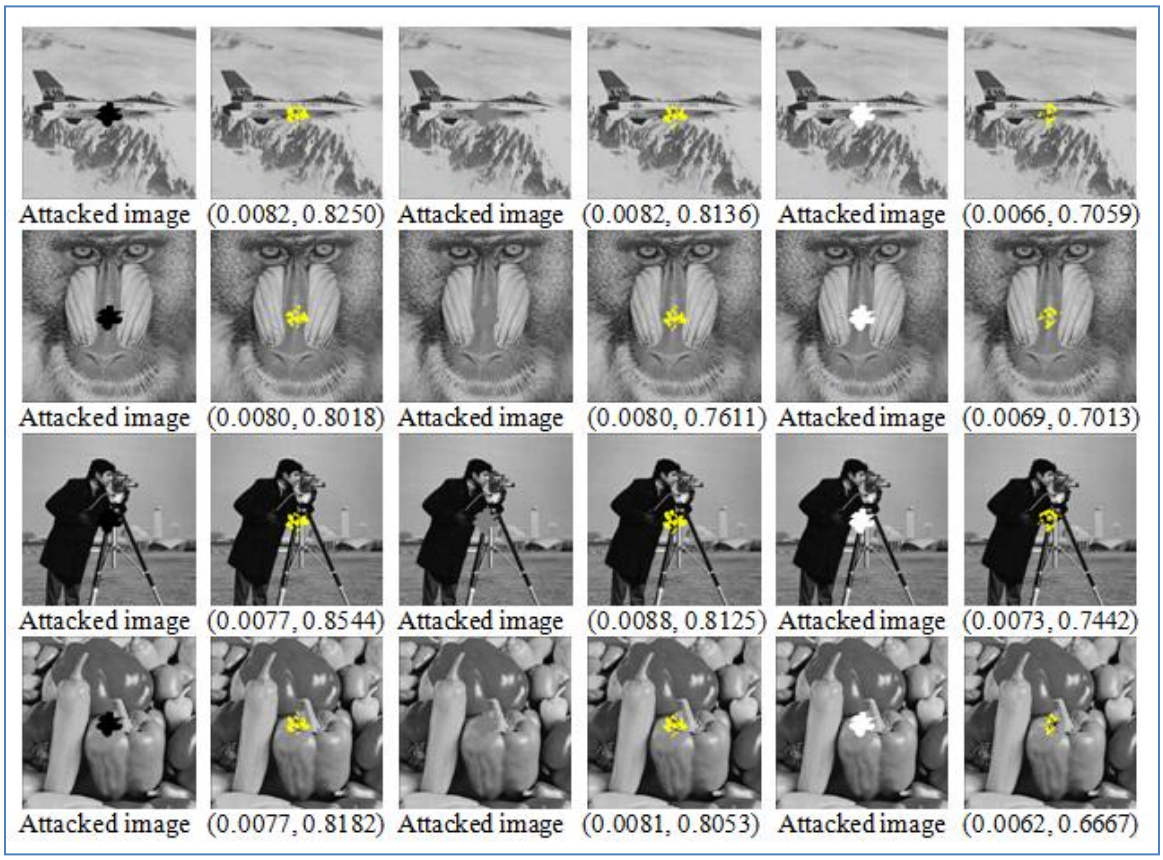


Figure 16. Illustration of the results of malicious attacks (irregular shape) of the proposed scheme.

result. This figure clearly shows that the proposed scheme successfully localizes the tampered regions. Based on the predefined thresholds, we conclude that the proposed scheme detects all these maliciously attacked watermarked images as maliciously tampered.

Figure 16 also illustrates the results of malicious attacks (irregular shape) of the proposed scheme: attacked images by adding black irregular shape (1st column), the corresponding detected distortion regions (2nd column), attacked images by adding gray irregular shape (3rd column), the corresponding detected distortion regions (4th column),

attacked images by adding white irregular shape (5^{th} column), and the corresponding detected distortion regions (6^{th} column).

We further applied three kinds of more realistic modifications on the watermarked "Lena" image by using Photoshop to insert an external object (decoration on hat), modify the right eye, and remove the object (white and gray wavy decoration) on the lower right, respectively. The maliciously attacked "Lena" image was then saved as a JPG image using the default compression setting. Figure 17 demonstrates the localization results, shown in yellow, of five schemes and lists the M_1 and M_2 values, whenever applicable, in a pair for each scheme. The figure clearly shows that the proposed scheme achieves the best and the cleanest localization results and that Maeno's scheme achieves the second best localization results with a few additional small isolated distorted regions resulting from the JPEG compression. Che's scheme achieves localization results comparable to Maeno's scheme except that it detects more distorted regions resulting from the JPEG compression due to less robustness to JPEG compression. Based on the thresholds for the two authentication measures, we conclude that the proposed scheme detects all three maliciously attacked watermarked "Lena" image as maliciously tampered and correctly localizes their tampered regions. Yang's scheme detects these maliciously attacked watermarked images as maliciously distorted. However, it does not produce a definite localization result under any of the three malicious attacks due to less robustness to JPEG compressions. The other three schemes obtain small values for M_1's, which are similar to the values obtained under image processing and JPEG compression attacks. As a result,

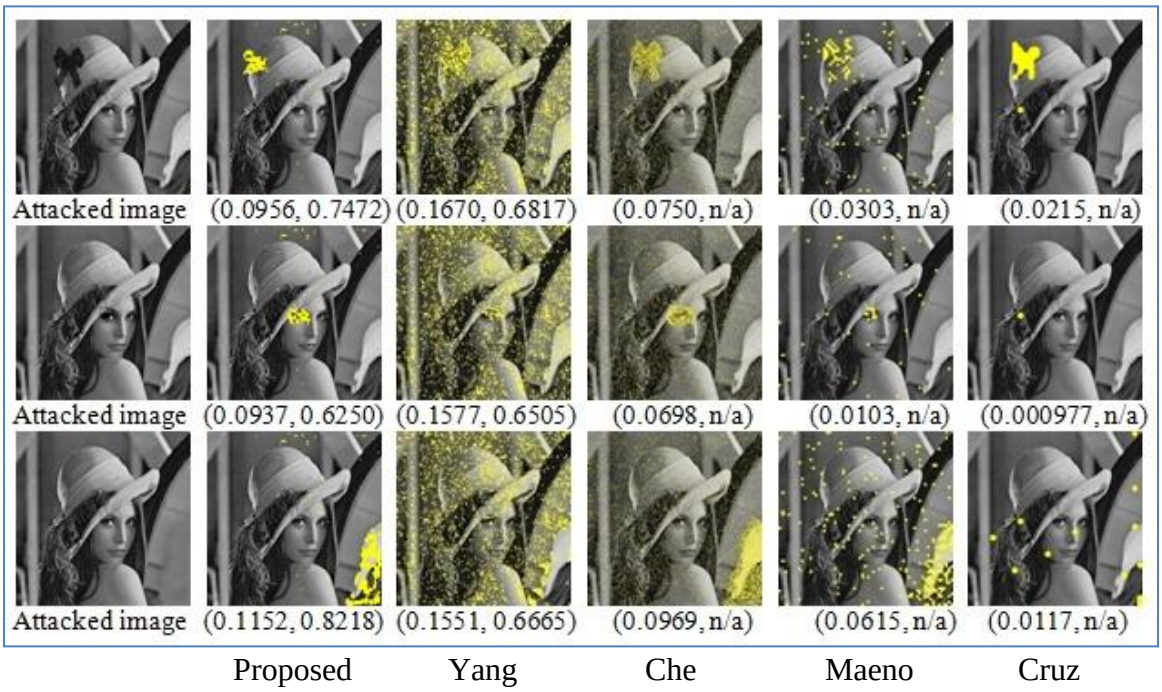


Figure 17. Comparison of malicious attacks (modified Lena by Photoshop).

they detect these maliciously attacked images as incidentally distorted based on the equivalent predefined thresholds.

Figure 17 shows the comparison of the localization results after realistic malicious attacks of the proposed scheme, Yang, Che, Maeno, and Cruz (from left to right).

Figure 18 demonstrates the tampering localization results on four additional representative watermarked images, which were maliciously attacked by inserting an external object or removing (modifying) an object using Photoshop. These maliciously attacked images were then saved as JPG images using the default compression setting. We also list the M_1 and M_2 values in a pair below each localization result. This figure clearly shows that the proposed scheme successfully localizes the tampered regions.

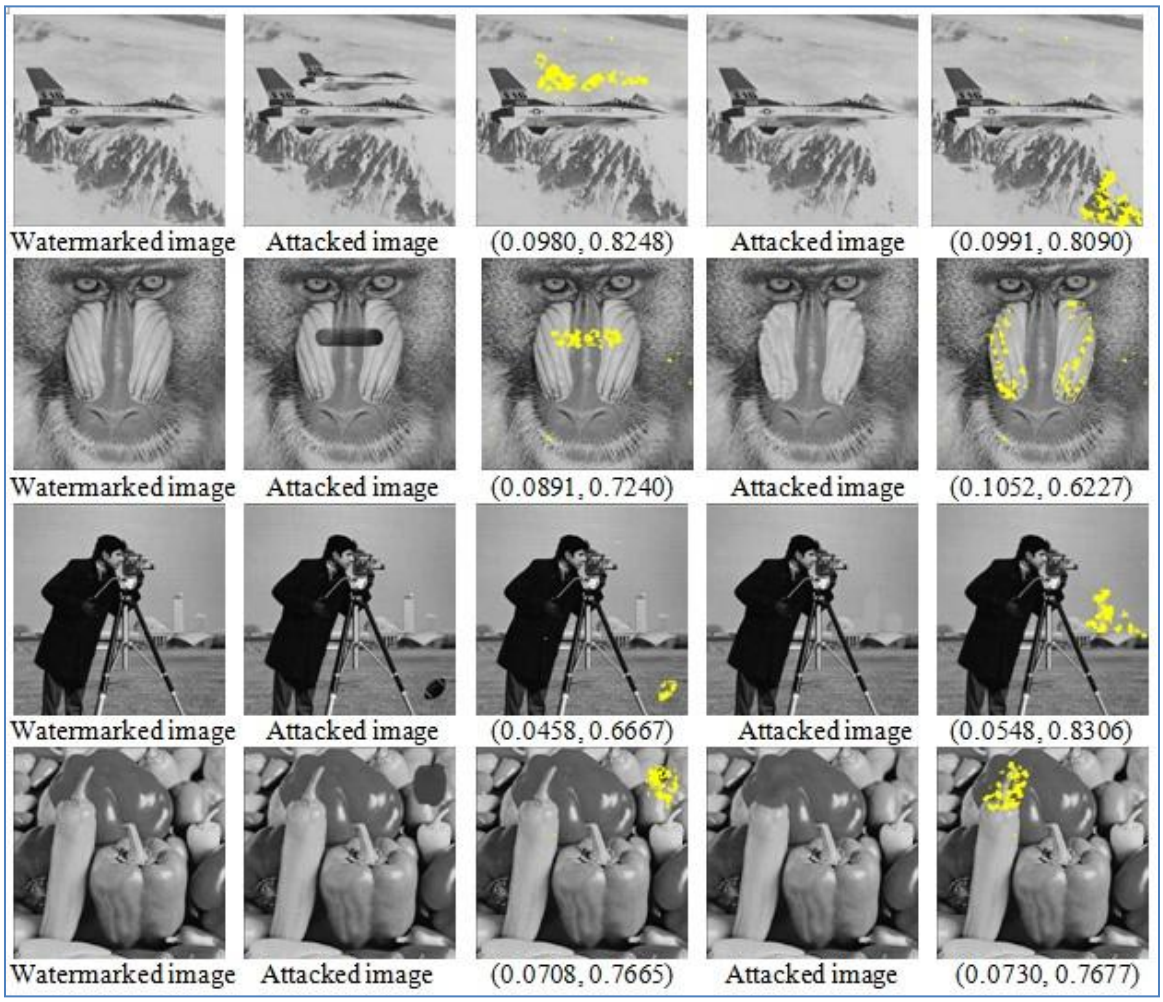


Figure 18. Illustration of the results of malicious attacks (modified by Photoshop) of the proposed scheme.

Based on the predefined thresholds, we conclude that the proposed scheme detects all these maliciously attacked watermarked images as maliciously tampered.

Figure 18 also illustrates the results of malicious attacks (modified by Photoshop) of the proposed scheme: watermarked images (1st column), maliciously attacked images by inserting an external object (2nd column), the corresponding detected distortion regions

(3rd column), maliciously attacked images by modifying or removing an object (4th column), and the corresponding detected distortion regions (5th column).

CHAPTER V

CONCLUSION AND FUTURE WORK

In this thesis, we present a novel semi-fragile watermarking scheme for image content authentication with tampering localization. The contributions of the proposed scheme are:

- Utilizing the three relationships of SV's of the horizontal, vertical, and diagonal subbands of each 4×4 block to extract content-based watermark in both watermark embedding and extraction processes.
- Utilizing the summation of the three relationships of SV's of the horizontal, vertical, and diagonal subbands of each 4×4 block to choose its adaptive quantizer q for both watermark embedding and extraction processes.
- Applying the quantization method to embed the secured watermark, which is obtained by applying the "*xor*" operation on content-based watermark and the private-key-based content-independent watermark, in the wavelet domain so that a majority of image distortions, which cause the intensity shift by a value larger than a half of the quantizer q, can be detected in the authentication process.
- Defining two authentication measures to quantitatively detect the authenticity of the probe image and prove tampering, with M_1 measuring the overall similarity between extracted and embedded watermarks and M_2 measuring the overall clustering level of the tampered error pixels.

- Using a binary error map together with the two authentication measures in the authentication process to compensate for possible misclassification in the error map, capture all possible distortions, and localize all possible tampered areas.

My extensive experimental results show that the proposed scheme successfully distinguishes malicious attacks from nonmalicious tampering of image content. It also accurately localizes maliciously tampered regions. My scheme is more robust to acceptable content-preserving operations and more fragile to malicious distortions than four semi-fragile watermarking schemes.

My future work includes studying the tampering detection sensibility of the proposed method when an image size changes, addressing geometric attack issues, and testing more images of various types.

REFERENCES

[1] Weiss, W. *Historische Wasserzerchen*. Saur, 1987.

[2] Liu, T. and Qiu, Z. The survey of digital watermarking-based image authentication techniques. In *Proc. of the 6th Int. Conf. on Signal Processing*, 2002, 1556-1559.

[3] Watermarker.com. 2004. http://www.watermarker.com/watermark-protector/watermark-examples.aspx Sep.2005.

[4] Hartung, F. and Kutter, M. Multimedia watermarking techniques. *IEEE Special Issue on Identification and Protection of Multimedia Information, 87*, 7 (1999), 1079-1107.

[5] Lin, E., Podilchun, C., and Delp, E. Detection of image alterations using semi-fragile watermarks. In *Proc. of SPIE Int. Conf. on Security and Watermarking of Multimedia Contents II*, 2000, 152-163.

[6] Lin, C. and Chang, S. Semi-fragile watermarking for authenticating JPEG visual content. In *Proc. of SPIE on Security and Watermarking of Multimedia Content II*, 2000, pp. 140-152.

[7] C. Ho and C. Li, Semi-fragile watermarking scheme for authentication of JPEG images. In *Proc. of Int. Conf. on ITCC*, 2004, pp.7-11.

[8] K. Maeno, Q. Sun, S. Chang, and M. Suto, New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization. *IEEE Trans. on Multimedia 8*, 1 (2006), 32-45.

[9] Kundur, D. and Hatzinakos, D. Digital watermarking for telltale tamper proofing and authentication. In *Proc. IEEE: Special Issue on Identification and Protection of Multimedia Information*, 1999, 1167−1180.

[10] Zhou, X., Duan, X., and Wang, D. A semi-fragile watermarking scheme for image authentication. In *Proc. of the 10th Int. Conf. on Multimedia Modeling*, 2004, 374.

[11] Kang, H. and Park, J. A semi-fragile watermarking using JND. In *Proc. of STEG*, 2003, 127−131.

[12] Hu, Y. and Han, D. Using two semi-fragile watermark for image authentication. In *Proc. of the 4th Int. Conf. on Machine Learning and Cybernetics*, 2005, 5484−5489.

[13] Liu, H., Lin, J., and Huang, J. Image authentication using content based watermark. In *Proc. of IEEE Int. Symp. on Circuits and Systems*, 2005, 4014−4017.

[14] Zhu, Y., Li, C., and Zhao, H. Structural digital signature and semi-fragile fingerprinting for image authentication in wavelet domain. In *Proc. of IAS*, 2007, 478–483.

[15] Yang, H. and Sun, X. Semi-fragile watermarking for image authentication and tamper detection using HVS model. In *Proc. of Int. Conf. on Multimedia and Ubiquitous Engineering*, 2007, 1112−1117.

[16] Che, S., Ma, B., and Che, Z. Semi-fragile image watermarking algorithm based on visual features. In *Proc. of Int. Conf. on Wavelet Analysis and Pattern Recognition*, 2007, 382-387.

[17] Cruz, C., Reyes, R., Nakano, M., and Perez, H. Image content authentication system based on semi-fragile watermarking. In *Proc. of 51st Midwest Symposium on Circuits and Systems*, 2008, 306-309.

[18] Wikipedia.com. JPEG. 2009. http://en.wikipedia.org/wiki/JPEG. Jan.2010

[19] Matsumoto, M. and Nishimura, T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. on Modeling and Computer Simulation 8*, 1 (1998), 3-30.

[20] Wikipedia.com. Parseval's theorem. http://en.wikipedia.org/wiki/Parsevel_theorem. Jan.2010

[21] Hsieh, M. and Tseng, D. Perceptual digital watermarking for image authentication in electronic commerce. *Electronic Commerce Research 4* (2004), 157-170.